Erweiterte Traceability zwischen Anforderungen und Design

Bernhard Turban, Markus Kucera, Athanassios Tsakpinis, Christian Wolff

Abstract

Traceability (Nachverfolgbarkeit) von Anforderungen wird zunehmend von den Entwicklungsprozessen für eingebettete Systeme gefordert. Der folgende Artikel zeigt drei Probleme, die eine Erlangung und Nutzung brauchbarer Traceability-Information zwischen Anforderungen und Design erschweren. Das Traceability-Werkzeug *PROVEtech®:R2A* bietet vor dem Hintergrund dieser Problemlage eine gezielte Unterstützung für einen kontrollierten Übergang von den Anforderungen zum Design und trägt dazu bei, Auswirkungen dieser Probleme zu minimieren.

1. Einleitung

Anforderungs-Traceability wird als wichtiger Faktor bei der Entwicklung von sicherheitskritischen Systemen wahrgenommen [SP08]. Die Gewinnung von Informationen zur Traceability stellt sich besonders im Übergang zwischen Anforderungen und Design als schwierig dar, da – anders als bisherige Ansätze suggerieren, die auf einfacher Verlinkung aufbauen – dieser Übergang nicht linear ist, sondern einen kreativen Transferprozess von einer Problemstellung zur Lösung darstellt, in dem die getroffenen Entscheidungen das Fundament bilden [TKT+07].

Die *MBtech Group* hat in Kooperation mit dem Kompetenzzentrum Software Engineering der Hochschule Regensburg und der Professur für Medieninformatik der Universität Regensburg das Traceability-Werkzeug *PROVEtech®:R2A* (R2A) entwickelt, das versucht, eine adäquate Lösung für diesen Übergang bereitzustellen. Die nachfolgenden Kapitel 2-4 beschreiben drei Probleme, die beim Übergang zwischen Anforderungen und Design die Etablierung der Traceability erschweren, und zeigen die Lösungsstrategien von R2A.

2. Design als eine Kette von Entscheidungen

Der Weg von Anforderungen zu dem sie realisierenden Design ist geprägt durch eine Abfolge Entscheidungen, die Lösungsraum immer weiter einengen. Dieser Umstand führt dazu, dass Design nicht nur von den Anforderungen, sondern zumeist in höherem Maße von den schon zuvor getroffenen Entscheidungen abhängt. Diese Beobachtung führt zu folgenden zwei Problemen:

- Entscheidungen und ihre Auswirkungen müssen im Projekt an andere Designer, Entwickler und Tester kommuniziert werden.
- Spätere Anforderungsänderungen beeinflussen nicht nur das Design, sondern können auch dazu führen, frühere Entscheidungen erneut prüfen und ggf. revidieren zu müssen, was wiederum auch Auswirkungen auf das Design haben kann.

Das Entwicklungstool R2A bietet derzeit zwei verschiedene Entscheidungsmechanismen, um Entscheidungen in ihren Beziehung zu Anforderungen als auch zum Design dokumentieren zu können.

2.1 Verwaltung begrenzter Ressourcen mittels Ressourcenbudgets

Projekte im Bereich *embedded systems* sind regelmäßig dadurch geprägt, dass Ressourcen wie RAM, ROM oder EEPROM nur sehr begrenzt verfügbar sind. R2A bietet die Möglichkeit, mittels eines hierarchischen Designs eingeschränkte Ressourcen im Sinne eines Budgets zu

verwalten und solche Festlegungen einzelnen Designelementen als „neue Anforderungen“ zuzuweisen. Dazu wurde für das Tool das Konzept eines *BudgetedResourceConstraint (BRC)*eingeführt. Abb. 1 zeigt wie eine RAM-Ressource von 1500 Bytes auf die SW-Architektur einer Lichtsteuerung verteilt wird.

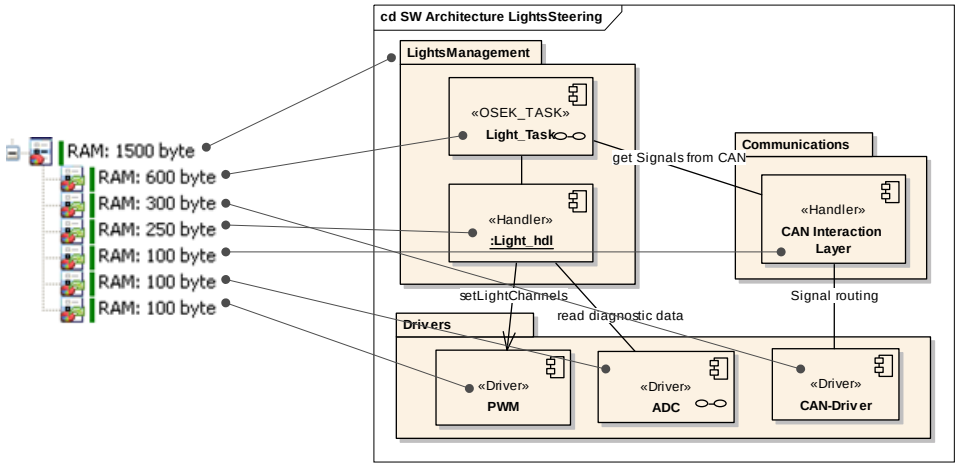

Abb. 1: Beispiel einer budgetbasierten Ressourcenverteilung auf ein Design.

Die so einem Designelement zugewiesenen *BRCs* gelten als weitere einzuhaltende Anforderungen und können entsprechend weiterverarbeitet werden. Abb. 2 zeigt wie ein anderer für das detaillierte Design Moduls *Light_hdl* zuständiger Designer eine *BRC* von 250 Bytes auf sein detaillierteres Design aufteilen kann. Weitere Informationen zu diesem Mechanismus finden sich in [TWT+08].

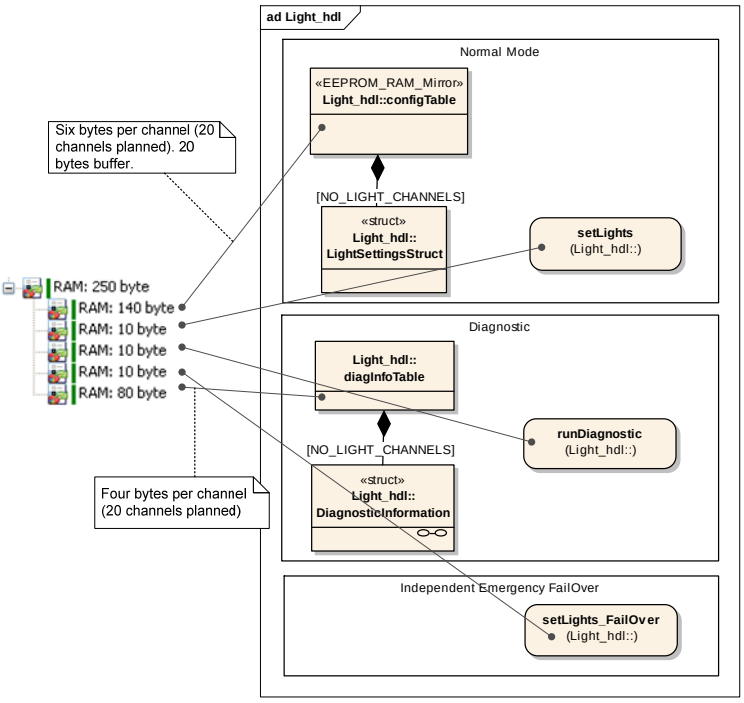

Abb. 2: Fortführung einer detaillierteren Ressourcenverteilung für Modul *Light_hdl*.

2.2 Modellierung von Konflikten zwischen Anforderungen und Design

Ein zweiter Mechanismus hilft, Konflikte und deren Lösung festzuhalten. Abb. 3 zeigt eine Situation, in der eine Anforderung („All 16 light ...“) in einem Konflikt mit dem PWM-Modul (Pulsweitenmodulation, PWM) steht, da die Hardware (HW) nur 12 PWM-Kanäle bietet. Als Lösung ergibt sich eine „neue Anforderung“ an das PWM-Modul. Diese „Anforderung“

stammt nicht aus den Kundenanforderungen, sondern aus der Designentscheidung und wird im Kontext des Tools R2A als *DesignConstraint* (*DC*) bezeichnet.

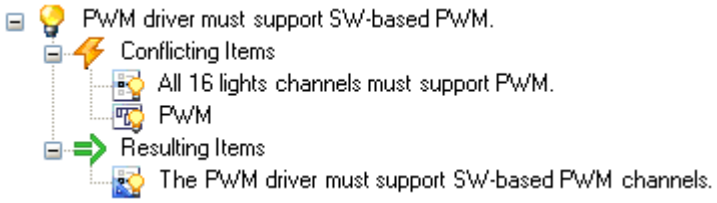

Abb. 3: Beispiel einer konfliktbasierten Entscheidung.

Mit den *DCs* bekommt der Designer die Möglichkeit, die Konsequenzen seiner Entscheidung an andere Designer zu kommunizieren. In diesem Beispiel wurde der neue *DC* dem PWM-Modul zugewiesen. Der für das detaillierte Design des PWM-Moduls zuständige Designer sieht in Zukunft diesen *DC* als ebenfalls zu erfüllende Anforderung, vgl. dazu detaillierter [TKT+07].

3. Behandlung von Anforderungsänderungen

Anforderungsänderungen treten in der Projektpraxis häufig auf und ihre Konsequenzen für den Entwicklungsprozess müssen genau nachverfolgt werden können. Das Werkzeug R2A bietet deshalb die Möglichkeit, Einflussanalysen (sog. Impact Analysen (IA)) durchzuführen, in denen grafisch auf verständliche Weise die Einflüsse von Anforderungsänderungen auch Außenstehenden (z.B. Kunden) kenntlich gemacht werden können. Abb. 4 zeigt wie Ergebnissituationen bei Einflussanalysen aussehen können. Direkte Auswirkungen sind dabei rot, indirekte gelb markiert. Links hat die Änderung nur sehr lokale Auswirkungen, während bei der rechten Situation direkte Einflüsse sowohl auf das ganze Design wie auch auf die Module PWM und OMT und viele indirekte Einflüsse sichtbar sind.

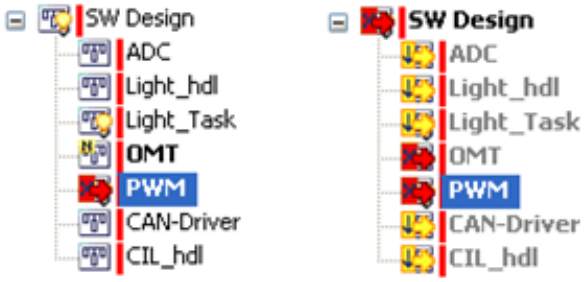

Abb. 4: Beispiele für die Visualisierung von Einflussanalysen

Neben direkten Einflüssen können sich auch aus den bereits getroffenen Entscheidungen Modifikationen für Designelemente ergeben. Abb. 5 zeigt eine geplante Änderung, die zunächst den *BRC* „RAM:250 byte“ und die *Anforderung* „All 16 light...“ in Betracht zieht. Daraus werden alle betroffenen Entscheidungen und davon abgeleitete *DCs* oder *BRCs* errechnet und unterhalb des Ausgangselements dargestellt. Auf der rechten Seite werden daraus folgende Konsequenzen auf das Design dargestellt: Rote Kreuze stehen für Einflüsse aus den Ausgangselementen, gelbe Kreuze für aus den Entscheidungen entstehende Einflüsse.

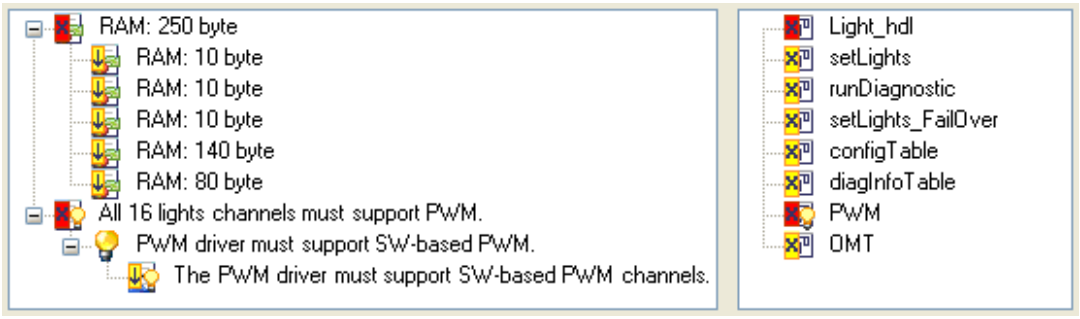

Abb. 5: Einflussanalyse unter Einbezug von Entscheidungen.

4. Wie kommen Anforderungsänderungen konsistent in das Design?

Wenn Anforderungsänderungen umgesetzt werden sollen, müssen die Änderungen kontrolliert in das Design überführt werden können, um die Konsistenz der Änderungen zu gewährleisten. R2A kann für jede Anforderung den aktuellen Status mit Hilfe eines farbigen Statusbalkens sichtbar machen. Dabei durchläuft jede Anforderung den in Abb.6 gezeigten Lebenszyklus. Jede nicht berücksichtigte Anforderung (rot) muss dem Design zugewiesen werden (gelb), um anschließend vom Designer umgesetzt zu werden (grün). Spätere Anforderungsänderungen können ein Überdenken der Umsetzung erforderlich machen (orange), bis der Designer die Änderung im Design wieder umgesetzt hat (erneut grün). Falls die Anforderung im Projektverlauf obsolet wird, wird sie als zu löschen markiert (grau). Sobald der Designer den Wegfall im Design berücksichtigt hat, wird sie endgültig gelöscht (schwarz).

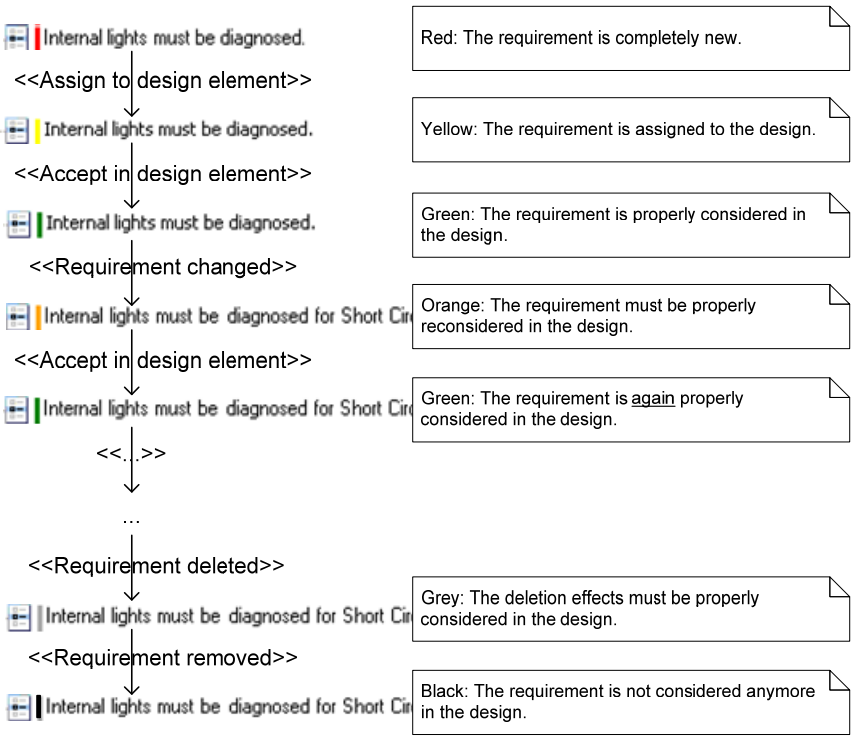

Abb. 6: Lebenszyklus einer Anforderung in R2A und ihre Farbkodierung

5. Zusammenfassung

Dieser Artikel illustriert drei Probleme, bei denen das im Requirements Engineering gängige Konzept einfacher Verlinkung von Anforderungen und Design in Verbindung mit der Verfolgung von Links als Operationalisierung der Nachverfolgbarkeit von Anforderungen (Traceability) wenig Unterstützung für die Architekten bietet. PROVEtech:R2A verfolgt das Ziel, Designern im Bereich *embedded systems* zu diesem Problemkomplex spezifische

Unterstützung bieten zu können. Das Tool PROVEtech:R2A wird voraussichtlich Anfang Februar 2009 als Produkt verfügbar sein.

Danksagung
Diese Entwicklung wird im Rahmen des FuE-Programm „Informations- und Kommunikationstechnik“ durch das Bayerische Staatsministerium für Wirtschaft, Infrastruktur, Verkehr und Technologie gefördert (Förderkennzeichen IUK229).

6. Literatur

[SP08] Schmied, J.; Palluch, J.*: Verschärfte Software-Qualität — mit Automotive SPICE*. http://www.elektronikpraxis.vogel.de/index.cfm?pid=904&pk=142646 (Zugriff 03.09.2008).

[TKT+07] Turban, B.; Kucera, M.; Tsakpinis, A.; Wolff, C.: *An Integrated Decision Model For Efficient Requirement Traceability in SPICE Compliant Development.* Fifth Workshop on Intelligent Solutions in Embedded Systems (WISES). Madrid 2007.

[TWT+08] Turban, B.; Wolff, C.; Tsakpinis, A.; Kucera, M.: *A Decision Model for Managing and Communicating Resource Restrictions in Embedded Systems Design.* Sixth Workshop on Intelligent Solutions in Embedded Systems (WISES). Regensburg 2008.

Bernhard Turban, ist Diplom-Informatiker (FH) und koordiniert in einem Forschungsprojekt der MBtech Group als technischer Projektleiter und Architekt die Entwicklung von PROVEtech:R2A. Gleichzeitig promoviert er an der Universität Regensburg über Konzepte der Nachverfolgbarkeit von Anforderungen.

Prof. Dr. Markus Kucera war nach seiner Promotion an der TU Wien sechs Jahre in der Automobilindustrie tätig und ist seit 2004 Professor für technische Informatik an der Hochschule Regensburg. Er ist Autor verschiedener Veröffentlichungen und zahlreicher Patente im Gebiet zuverlässige Systeme.

Prof. Dr. Athanassios Tsakpinis lehrt Wirtschafts- und Medizininformatik an der Hochschule Regensburg. Seine Lehrgebiete sind Software Engineering, ERP-Systeme und Krankenhausinformationssysteme. Das von ihm geleitete Kompetenzzentrum Software Engineering nimmt an Industrieprojekten in diesen Lehrgebieten teil.

Prof. Dr. Christian Wolff ist Informationswissenschaftler und Informatiker und seit 2003 Professor für Medieninformatik an der Universität Regensburg. Seine Forschungsgebiete sind Information Retrieval, multimediale und multimodale Informationssysteme sowie Software und Usability Engineering.